



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|--|-------------|----------------------|--------------------------------|------------------|
| 09/870,624 | 05/31/2001 | Scott J. Broussard | AUS920010268US1 | 1775 |
| 35617 | 7590 | 02/22/2005 | | |
| DAFFER MCDANEIL LLP P.O. BOX 684908 AUSTIN, TX 78768 | | | EXAMINER BONSHOCK, DENNIS G | |
| | | | ART UNIT 2173 | PAPER NUMBER |

DATE MAILED: 02/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/870,624

Applicant(s)

BROUSSARD, SCOTT J.

Examiner

Dennis G. Bonshock

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 November 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: _____.

Art Unit: 2173

Final Rejection

1. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment as received on 11-08-04.
2. Claims 1-20 have been examined.

Status of Claims:

3. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nason, Patent #6,727,918, and Fowler, "Mixing heavy and light components."

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nason, Patent #6,727,918, and Fowler, "Mixing heavy and light components."
3. With regard to claim 1, Nason teaches a display (see column 5, lines 33-44), a graphical user interface (see column 5, lines 18-21), a processor for implementing the embodiments of the invention (see claim 48), the system being implemented on a windows based operating system (see column 2, lines 46-50), implementing two different APIs, one independent of OS, one dependent on OS, to generate images (see column 5, lines 18-22 and lines 45-63), and a reading

Art Unit: 2173

and rewriting of screen display information, where the primary GUI information is maintained by replacing the primary GUI with a secondary GUI (see column 25, lines 27-40). Nason also teaches the use of Java, but doesn't get in to the specifics. Fowler teaches a system of mixing two APIs similar to that of Nason (see page 1, paragraphs 1 and 2), but further teaches specifics of Java's interfaces AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). Fowler further teaches, on page 7, the components having the same look and feel. It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before him at the time the invention was made to modify the system of using two APIs, of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs in Java, which is taught by Nason. Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Nason.

4. With regard to claim 2, which teaches the first and second images having the same look and feel, Fowler further teaches, on page 7, the components having the same look and feel.

Art Unit: 2173

5. With regard to claims 3 and 11, which teach the first and second images comprising pixels presented upon the display via the graphical user interface associated with the application program, Nason further teaches, in column 6, lines 14-37, the use of pixels for presenting the image on the display.

6. With regard to claims 4 and 12, which teach the first and second images comprise images of an object selected from a group comprising buttons, list boxes, and slide bars on which a pointer device can be directed by a user, Nason teaches, in column 6, lines 25-30, the display not being limited to, buttons, menus, application output controls animations, and use input controls. Furthermore, Fowler teaches, on page 1, paragraph 1, Swing an AWT containing components such as buttons, lists, and the like.

7. With regard to claims 5 and 13, which teach an application program written in Java programming language, Nason further teaches, in column 5, lines 60-63, the content controller including content and operating software such as JAVA.

8. With regard to claims 6 and 14, which teach the software component comprising a java application program interface consisting of an abstract windowing toolkit (AWT) during a second time, Nason further teaches, the use Java in the system. Fowler teaches a system in which two APIs can be used, similar to that of Nason, but also teaches the use of Swing and AWT in the same application program (see page 1, paragraph 2). It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before him at the time the invention was made to modify the system of using two APIs,

Art Unit: 2173

of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs used in Java, which is also mentioned in Nason (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Nason.

9. With regard to claims 7 and 15, which teach the software component comprising a java application program interface consisting of an Swing application program interface during a first time, Nason further teaches, the use Java in the system. Fowler teaches a system in which two APIs can be used, similar to that of Nason, but also teaches the user of Swing and AWT in the same application program (see page 1, paragraph 2). It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before him at the time the invention was made to modify the system of using two APIs, of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs used in Java, which is also mentioned in Nason (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Nason.

Art Unit: 2173

10. With regard to claims 8 and 16, which teach the operating system comprising a Windows, Unix, or OS/2 computer operating system, Nason further teaches, in column 2, lines 46-50, the use of operating systems such as Windows, Linux, Apple's Macintosh OS/2, or Unix.

11. With regard to claims 9 and 17, which teach the first and second images presenting the same look and feel upon the display independent of the operating system, Fowler further teaches, on page 2, that the java look and feel, that of AWT and Swing, that provides a distinctive platform-independent look and feel.

12. With regard to claim 10, Nason teaches implementing two different APIs, one independent of OS and one dependent on OS, to generate images (see column 5, lines 18-22 and lines 45-63), and a reading and rewriting of screen display information, where the primary GUI information is maintained by replacing the primary GUI with a secondary GUI (see column 25, lines 27-40). Nason also teaches the use of Java, but doesn't get in to the specifics. Fowler teaches a system of mixing two APIs similar to that of Nason (see page 1, paragraphs 1 and 2), but further teaches specifics of Java's interfaces AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). Fowler further teaches, on page 7, the components having the same look and feel. It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before

Art Unit: 2173

him at the time the invention was made to modify the system of using two APIs, of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs in Java, which is taught by Nason. Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Nason.

13. With regard to claim 18, Nason teaches a computer-readable storage device, comprising: an operating system (see column 2, lines 46-50), and application program adapted for executing code of a software component (ex: APIs) (see column 3, lines 36-39 and column 5, lines 45-64), implementing two different APIs to generate objects in the same application program (see column 5, lines 45-64), and a reading and rewriting of screen display information, where the primary GUI information is maintained by replacing the primary GUI with a secondary GUI (see column 25, lines 27-40). Nason also teaches the use of two Java APIs, but doesn't get in to the specifics of each of the APIs. Fowler teaches a system of mixing two APIs similar to that of Nason (see page 1, paragraphs 1 and 2), but further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). Fowler further teaches, on page 7, the components having

Art Unit: 2173

the same look and feel. It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before him at the time the invention was made that the first image would overwrite an image upon the display previous to the first image and that the first image can't overwrite the second image during the first time (because it hasn't been generated yet) and also to modify the system of using two APIs in one application program, of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because the APIs of Java are use in Nason. Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Nason.

14. With regard to claim 19, which teach the software component comprising a java application program interface consisting of a Swing application program interface during a first time, Nason further teaches, the use Java in the system for complementary user interfaces coexisting, with a primary interface. Fowler teaches a system in which two APIs can be used, similar to that of Nason, but also teaches the user of Swing and AWT in the same application program (see page 1, paragraph 2). It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before him at the time the invention was made to modify the system of using two APIs, of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs used in Java, which is also mentioned in Nason (see column 1, line

Art Unit: 2173

39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Nason.

15. With regard to claim 20, which teaches the software component comprising a java application program interface consisting of an abstract windowing toolkit (AWT) during a second time, Nason further teaches, the use Java in the system for complementary user interfaces coexisting, with a primary interface. Fowler teaches a system in which two APIs can be used in the same application program similar to that of Nason, but also teaches the use of Swing and AWT in the same application program (see page 1, paragraph 2). Fowler further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Nason and Fowler before him at the time the invention was made to modify the system of using two APIs, of Nason, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs implemented in Java, which is claimed Nason. Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1,

Art Unit: 2173

paragraph 2), would add an element of platform independence to the invention of Nason.

Response to Arguments

16. The arguments filed on 11-08-2004 have been fully considered but they are not persuasive. Reasons set forth below.

17. The applicants' argue that Nason definitely fails to teach or suggest that an image (within, e.g., a complementary GUI) may be displayed with a platform-independent interface during a first time, while during a second time, the same image may be displayed with a platform-specific interface.

18. In response, the examiner respectfully submits that Nason teaches, in column 5, lines 18-22 and lines 45-63, implementing two different APIs, one independent of OS, one dependent on OS, to generate images. Nason further teaches, in column 25, lines 27-40, a reading and rewriting of screen display information, where the primary GUI information is maintained by replacing the primary GUI with a secondary GUI.

19. The applicants' argue that Nason also fails to provide teaching or suggestion for an application program that generates an OS-dependent image during a first time and an OS-independent image during a second time, where the first and second images are substantially identical and the second image is adapted to overwrite the first image upon a display screen.

20. In response, the examiner respectfully submits that Nason teaches, in column 5, lines 18-22 and lines 45-63, implementing two different APIs, one independent of OS, one dependent on OS, to generate images. Nason further

Art Unit: 2173

teaches, in column 25, lines 27-40, a reading and rewriting of screen display information, where the primary GUI information is maintained by replacing the primary GUI with a secondary GUI. Fowler further teaches, that the Java API AWT uses heavyweight components and the Java API Swing uses lightweight components (see page 2, paragraphs 1 and 3), and that the lightweight components and heavyweight components look substantially identical (see page 7).

21. The applicants' argue that Fowler teaches away from displaying an image with the AWT API during a first time, and displaying the same image with the Swing API during a second time.

22. In response, the examiner respectfully submits that though Fowler warns of problems that may be encountered when mixing AWT and Swing, Fowler teaches, on page 1, paragraph 2, the mixing of AWT and Swing in the same application program, where the teaching of display of the same image with a second API is taught by Nason, in column 25, lines 27-40, as shown supra.

Art Unit: 2173

Conclusion

23. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

24. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.


25. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis G. Bonshock whose telephone number is (571) 272-4047. The examiner can normally be reached on Monday - Friday, 6:30 a.m. - 4:00 p.m.

26. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2173

27. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

2-18-05
dgb



RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173